

On Optimization of Expertise Matching with Various Constraints

Wenbin Tang^a, Jie Tang^a, Tao Lei^a, Chenhao Tan^b, Bo Gao^a, Tian Li^c

^aDepartment of Computer Science and Technology, Tsinghua University, Beijing 10084, China

^bDepartment of Computer Science, Cornell University, 4121 Upson Hall, Ithaca, NY 14850, USA

^cBeijing University of Aeronautics and Astronautics, China

Abstract

This paper studies the problem of expertise matching with various constraints. Expertise matching, which aims to find the alignment between experts and queries, is a common problem in many applications such as conference paper-reviewer assignment, product-reviewer alignment, and product-endorser matching. Most existing methods formalize this problem as an information retrieval problem and focus on finding a set of experts for each query independently. However, in real-world systems, various constraints are often needed to be considered. For example, in order to review a paper, it is desirable that there is at least one senior reviewer to guide the reviewing process. An important question is: “Can we design a framework to *efficiently* find the *optimal solution* for expertise matching under various constraints?” This paper explores such an approach by formulating the expertise matching problem in a constrain-based optimization framework. In the proposed framework, the problem of expertise matching is linked to a convex cost flow problem, which guarantees an optimal solution under various constraints. We also present an online matching algorithm to support incorporating user feedbacks in real time. The proposed approach has been evaluated on two different genres of expertise matching problems, namely conference paper-reviewer assignment and teacher-course assignment. Experimental results validate the effectiveness of the proposed approach. Based on the proposed method, we have also developed an online system for paper-reviewer suggestions, which has been used for paper-reviewer assignment in a top conference and feedbacks from the conference organizers are very positive.

Keywords: Expertise Matching, Constrained Optimization, Paper-Reviewer Assignment, Topic Models

1. Introduction

The fusion of computer technology and human collective intelligence has recently emerged as a popular way for users to find and share information on the internet. For example, ChaCha.com, one of the largest mobile search engines, has already attracted users to answer over 300 million questions; Epinions.com, a consumer review site, has already collected thousands of millions of reviews for products. The human-based computation offers a new direction in search with its unique use of human intelligence; however, it also poses some brand new challenges. One key problem, referred to as expertise matching, is how to align human experts with questions (queries)? Straightforwardly, we hope that the human experts who are assigned to answer a question have the specific expertise related to the question. But it is obviously insufficient. An ideal matching system should also consider various constraints in the real world, for example, an expert can only answer a certain number of questions (load balance); as the authoritative degree of different experts may vary largely, it is desirable that each question can be answered/reviewed by at least one senior expert (authority balance); a question may be relevant to multiple different aspects (topics), thus it is expected that the combined expertise of all assigned experts could cover all aspects of questions (topic coverage).

The problem has attracted considerable interest from different domains. For example, several works have been made for

conference paper-reviewer assignment by using methods such as mining the web [1], latent semantic indexing [2], probabilistic topic modeling [3][4], integer linear programming [5], minimum cost flow [6] and hybrid approach of domain knowledge and matching model[7]. A few systems such as [8][9][10] have also developed to help proposal-reviewer and paper-reviewer assignment. However, most existing methods mainly focus on improving the accuracy of measuring the relevance between queries and experts, i.e., how to find (or rank) relevant experts for each query, but ignore the different constraints or tackle the constraints using heuristics. Moreover, these methods usually do not consider user feedbacks. On the other hand, there are some methods focusing on expert finding. For example, Fang et al. [11] proposed a probabilistic model for expert finding, and Petkova et al. [12] employed a hierarchical language model in enterprise corpora. Balog et al. [13] employ probabilistic models to study the problem of expert finding, which tries to identify a list of experts for a query. However, these methods retrieve experts for each query independently, and cannot be directly used to deal with the expertise matching problem. Thus, several key questions arise for expertise matching, i.e., how to design a framework for expertise matching to guarantee an optimal solution under various constraints? how to develop an online algorithm so that it can incorporate user feedbacks in real time?

Figure 1 shows an example of paper-reviewer matching (as-

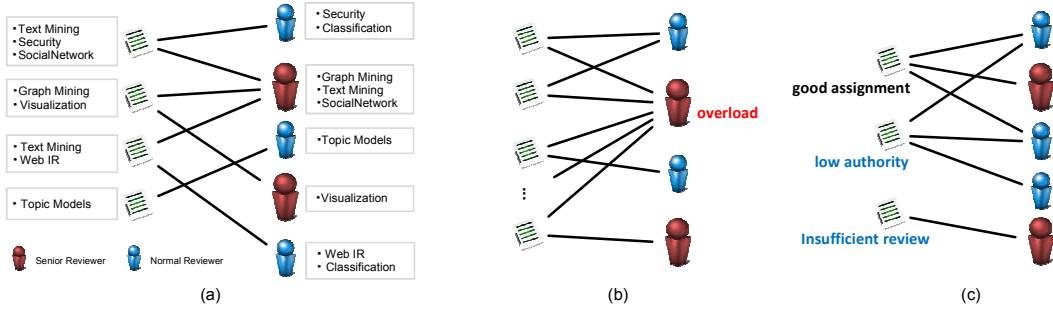


Figure 1: Possible constraints in the paper-reviewer matching example: (a) each paper has one or several topics, each reviewer has related expertise, and the assigned reviewer should cover some topics of the paper; (b) each reviewer can only review a certain number of papers; and (c) a good assignment is the paper reviewed by a group of reviewers, including at least one senior reviewer.

signing reviewers to each paper). In the paper-reviewer matching problem, the topics corresponding to a paper (also the expertise of the reviewers) can be “machine learning”, “data mining”, “computational theory”, etc. Also, each paper has a distribution on different topics. First, for each paper, the assigned reviewers’ expertise should cover the topics of the paper, and all the reviewers should have a load balance (each reviewer can only review a certain number of papers). In addition, some reviewers might be senior and some might be junior. We always hope that the review of each paper can be “supervised” by a senior reviewer. Another example is the patient-doctor matching case, the topics corresponding to the doctor’s expertise include “pediatrics”, “rheumatology”, “neurology”, etc. Each doctor has different expertise scores on different topics, while the disease of a patient also has a relevance distribution on the topics. Ideally, when arranging a consultation for a patient, the topics of the assigned doctors should contain the potential causes of the patient’s disease (e.g. a consultation for an SLE patient requires rheumatologists, nephrologists, cardiologists, neurologists), and all the doctors should have a load balance so that no doctor overstrains.

Contributions In this paper, we formally define the problem of expertise matching and propose a constrain-based optimization framework to solve the problem. Specifically, the expertise matching problem is cast as a convex cost flow problem and the objective is then to find a feasible flow with minimum cost under certain constraints. We theoretically prove that the proposed framework can achieve an optimal solution under various constraints and develop an efficient algorithm to solve it. By extending our preliminary work [14], we re-formalize our framework considering “multi-topic coverage” matching, which is very important to paper-reviewer assignment problem. We show that topic coverage measures (e.g. *Coverage* and *Confidence*) [3] can also be added as constraints to our framework. To evaluate whether our approach can achieve better multi-topic coverage, we compared our framework with several state of the art approaches on an additional dataset. Experimental results substantiate the effectiveness and efficiency of the proposed approach. We have applied the proposed method to help assign reviewers to papers for a top conference. Feedbacks from the conference organizers confirm the usefulness of

the proposed approach.

The rest of the paper is organized as follows: Section 2 reviews the relevant literatures; Section 3 formally formulates the problem; Section 4 explains the proposed optimization framework. Section 5 gives experimental results that validate the effectiveness and the computational efficiency of our methodology. Finally, Section 6 concludes.

2. Related Work

In general, existing methods for expertise matching mainly fall into two categories: probabilistic model and optimization model. The probabilistic model tries to improve the matching accuracy between experts and queries based on different probabilistic models such as keyword matching [1], latent semantic indexing [2], probabilistic topic modeling [3][4]. However, most of these methods do not consider the various constraints or simply consider the constraints by heuristics. The optimization model tries to incorporate the constraints as a component in an optimization framework such as integer linear programming [5], minimum cost flow [6].

Most previous works cast expert matching or expert finding as an information retrieval problem, in which every expert is represented as a “expertise” document and given a query, the goal is to retrieve most relevant experts. As a result, these methods mainly focus on two points: how to define the matching score between a query and a document; and how to represent each expert [15][16]. For example, Dumais and Nielsen [2] use Latent Semantic Indexing (LSI) as the retrieval method, and abstracts provided by reviewer as expertise documents. Yu et al. [17] represent experts by analyzing text content and extracting related information. Basu et al. [1][18][19] integrate different sources of information for recommendation (e.g. publications, research interests, etc.). Yarowsky and Florian [20] assign a paper by computing its cosine similarity with a reviewer and choosing the one with the highest rank.

In addition, different language models [11][12][13][21][22] and topic models [23] are used for expert matching/finding problem. In all of the language models, the matching score is the probability of a query given an expertise document i.e. $p(q|d)$, but its definition varies. Mimno and McCallum [4] improve the matching accuracy by proposing a novel topic model

Table 1: Notations.

| SYMBOL | DESCRIPTION |
|------------------|---|
| M | number of experts |
| N | number of queries |
| T | number of topics |
| V | the set of candidate experts |
| Q | the set of queries |
| v_i | one expert |
| q_j | one query |
| $\theta_{v_i z}$ | the probability of topic z given expert v_i |
| $\theta_{q_j z}$ | the probability of topic z given query q_j |
| $T(v_i)$ | the set of major related topics of expert v_i |
| $T(q_j)$ | the set of major related topics of query q_j |

Author-Persona-Topic (APT), in which experts are represented as independent distributions over topics. Karimzadehgan et al. also consider matching experts on multiple aspects of expertise [3]. Unlike the previous probabilistic models in which a query is matched as a whole unit, it tries to find “comprehensive” matchings to cover all subtopics of a query. New measures *Coverage* and *Confidence* are defined to evaluate multi-aspect expertise matching results. Several matching methods are proposed and show better multi-aspect performance than traditional ones.

However most of the aforementioned works treat each query independently and ignore the certain constraints (e.g. load balance), thus they cannot be directly adapted to an expertise matching system. In the real world, expertise matching is a highly constrained problem, and some existing works study this constrained optimization problem using various methods. For example, Guervós et al. [24] combine a greedy and an evolutionary algorithm [25][26] to assign papers to reviewers. Karimzadehgan et al. [5] and Taylor [27] cast it as an Integer Linear Programming (ILP) problem so approximate solutions can be found by any ILP solver. Recently, a few systems [8][9][10][28][29] have also been developed to help proposal-reviewer and paper-reviewer assignment. However, the expertise matching problem is still treated as an information retrieval problem, which obviously cannot result in an optimal solution.

In this paper, we aim to formalize the problem of expertise matching in a constrain-based optimization framework and propose an efficient algorithm to solve the framework. The differences of our work from existing work are: (a) we offer an optimization framework that incorporates the expertise matching and various constraints together; (b) the framework can be easily extended since new constraint can be combined into the optimization framework by simply defining a new (hard or soft) constraint; and (c) the framework can guarantee an optimal solution.

3. Problem Formulation

In this section, we first give several necessary definitions and then present a formal definition of the problem.

Given a set of experts $V = \{v_i\}$, each expert has different expertise over all topics. Formally, we assume that there are

T aspects of expertise (called topics) and each expert v_i has different expertise degrees on different topics. Further, given a set of queries $Q = \{q_j\}$, each query is also related to multiple topics. Given this, we first define the concept of topic model.

Definition 1. Topic model. A topic model θ of an expert (or a query) is a multinomial distribution of words $\{p(w|\theta)\}$. Each expert (query) is considered as a mixture of multiple topic models. The assumption of this model is that words associated with the expert (query) are sampled according to the word distributions corresponding to each topic, i.e., $p(w|\theta)$. Therefore, words with the highest probability in the distribution would suggest the semantics represented by the topic.

Assuming we have T topics, the expertise degree of expert v_i on topic $z \in \{1 \dots T\}$ is represented as a probability $\theta_{v_i z}$ with $\sum_z \theta_{v_i z} = 1$. Similarly, for each query, we also have a T -dimensional topic distribution with $\sum_z \theta_{q_j z} = 1$. Notations are summarized in Table 1.

It is easy to understand that each query q can be represented as a sequence of words, i.e., \mathbf{w}_q . To represent every expert v_i , without loss of generality, we also consider it as a sequence of words, i.e., \mathbf{w}_{v_i} . Based on this representation, we can calculate the similarity (or relevance score) between each query and every expert using measures such as cosine similarity or language model. Given this, we can define our problem of expertise matching with various constraints.

Problem 1. Expertise matching with constraints. Given a set of experts V and a set of queries Q , the objective is to assign m experts to each query by satisfying certain constraints, such as (1) m should be in a range $[n_1, n_2]$, where $n_1 \leq n_2$; (2) the experts’ topic model should cover the query’s topic model; (3) the assignment should avoid some conflict-of-interest (COI).

Actually in some applications, satisfying the constraints is more important than matching expertise with the queries. For example, in the conference paper-reviewer assignment, the authors of a paper should not be assigned to review their own paper. This must be a hard constraint. While in some other scenario, the constraint is relatively soft, for example the load balance of each expert. The number of assigned queries to each expert can be in a range between n_1 and n_2 . In existing works, Dumais et al. [2] and Mimno et al. [4] mainly focus on improving the accuracy of expertise matching, but ignore how to obtain an optimal matching satisfying the various constraints. Karimzadehgan et al. [5] use integer linear programming to find the solution for expertise matching with constraints. However, the proposed model cannot guarantee an optimal solution. In this work, we propose a generalizable optimization framework to solve this problem. Various constraints can be easily incorporated in the framework.

4. The Constrained Optimization Framework

In this section, we propose a constrain-based optimization framework for expertise matching. We develop an efficient algorithm to solve the optimization framework based on the theory of convex-cost flow, and also present an online matching

algorithm to incorporate user feedbacks into the framework in real time.

Basic Idea. The basic idea of our approach is to formulate this problem in a constrained optimization framework. Different constraints can be formalized as penalty in the objective function or be directly taken as the constraints in the optimization solving process. For solving the optimization framework, we transform the problem to a convex cost network flow problem, and present an efficient algorithm which guarantees an optimal solution.

4.1. The Framework

Now, we explain the proposed approach in detail. In general, our objective can be viewed from two perspectives: maximizing the matching score between experts and queries and satisfying the given constraints. Formally, we denote the set of experts to answer query q_j as $V(q_j)$, and the set of queries assigned to expert v_i as $Q(v_i)$. Further, we denote the matching score between expert v_i and query q_j as R_{ij} . Therefore, a basic objective function can be defined as follows

$$\text{Max} \sum_{v_i \in V} \sum_{q_j \in Q(v_i)} R_{ij} \quad (1)$$

The objective function can be equivalently defined as $\sum_{q_j \in Q} \sum_{v_i \in V(q_j)} R_{ij}$. In different applications, the constraints will be different. Here we use several general constraints to explain how the proposed framework can incorporate different constraints.

The first constraint is that each query should be assigned to exactly m experts. For example, in the paper-reviewer assignment task, each paper should be assigned to 3 or 5 reviewers. This constraint can be directly added into the optimization problem. Formally, we have:

$$\text{ST1} : \forall q_j \in Q, |V(q_j)| = m \quad (2)$$

The second constraint is called as *expert load balance*, indicating that each expert can only answer a limited number of queries. There are two ways to achieve this purpose: define a *strict* constraint or add a *soft penalty* to the objective function. For *strict*, we add a constraint indicating that the number of assigned queries to every expert v_i should be equal or larger than a minimum number n_1 , but be equal or smaller than a maximum number n_2 . The *strict* constraint can be written as:

$$\text{ST2 (strict)}: \forall v_i \in V, n_1 \leq |Q(v_i)| \leq n_2 \quad (3)$$

The other way is to add a soft penalty to the objective function (Eq. 1). For example, we can define a square penalty as $|Q(v_i)|^2$. By minimizing the sum of the penalty $\sum_i |Q(v_i)|^2$, we can achieve a *soft* load balance among all experts, i.e.:

$$\text{soft penalty: Min} \sum_{v_i \in V} |Q(v_i)|^2 \quad (4)$$

These two methods can be also used together. Actually, in our experiments, soft penalty method gives better results than strict constraint. Combining them together can yield a further improvement.

The third constraint is called *authority balance*. In real application, experts have different expertise level (authoritative

level). Take the paper-reviewer assignment problem as an example. Reviewers may be divided into 2 levels: senior reviewers and average reviewers. Intuitively, we do not hope that the assigned reviewers to a paper are all average reviewers. It is desirable that the senior reviewers can cover all papers to guide (or supervise) the review process. Without loss of generality, we divide all experts into K levels, i.e., $V^1 \cup V^2 \cup \dots \cup V^K = V$, with V^1 representing experts of the highest authoritative level. Similar to *expert load balance*, we can define a strict constraint like $|V^1 \cap V(q_j)| \geq 1$, and also add a penalty function to each query q_j over the k -level experts. Following, we give a simple method to instantiate the penalty function:

$$\text{Min} \sum_{k=1}^K \sum_{j=1}^N |V^k \cap V(q_j)|^2 \quad (5)$$

Besides the above constraints, sometimes we wish to assign experts that can cover all topics in the query. In the reviewer assignment problem, for example, a paper may be related to several research areas, thus ideally a comprehensive assignment is the paper reviewed by experts that cover all of them.

We define related topics of a query $T(q_j)$ and subject areas of an expert $T(v_i)$ (i.e. topics which the expert is good at). $T(q_j)$ and $T(v_i)$ can be determined by different ways. For example, authors may select the related categories and subjects when submitting their paper, and reviewers have to choose their primary and secondary subject before assigning papers to them. In addition, it is also possible to estimate their topics by the learned topic distributions: (a) we select top- k topics in $\theta_{v_i}, \theta_{q_j}$ as their related topics; or (b) we use thresholds τ_v and τ_q to prune topics so that related topics satisfy $\theta_{v_i} > \tau_v$ and $\theta_{q_j} > \tau_q$.

Follow previous work [3] we can incorporate different evaluation measures for topic covering. One measure is called *Coverage*, as we hope assigned experts can cover different distinct topics of a given query, i.e.,

$$\text{Coverage}(q_j) = \frac{|T(q_j) \cap (\bigcup_{v_i \in V(q_j)} T(v_i))|}{|T(q_j)|} \quad (6)$$

In this way, an optimal assignment should maximize the *Coverage*, i.e. $T(q_j) \subseteq \bigcup_{v_i \in V(q_j)} T(v_i)$, but this becomes the NP-hard set cover problem which is intractable to find a best answer. Consequently, we choose to find less optimal solutions by making further assumptions. We suppose each expert $v_i \in V(q_j)$ can select only one major topic $\hat{T}_{q_j}(v_i) \in T(v_i)$, and he is responsible to cover this topic for the given query. Finding the matching of responsible topics to assigned experts which maximizes the coverage actually opens another optimization problem, but fortunately this can be transformed into a network flow problem as well, which will be discussed in Section 5.2.

Another measure is called *Confidence*, as we prefer the related topic be covered by as many experts as possible, i.e.,

$$\text{Confidence}(q_j) = \frac{1}{m} \sum_{v_i \in V(q_j)} \frac{|T(q_j) \cap T(v_i)|}{|T(q_j) \cap (\bigcup_{v_i \in V(q_j)} T(v_i))|} \quad (7)$$

Generally there is a *Coverage – Confidence* tradeoff. To achieve both high coverage and high confidence, a measure *Average Confidence* is used to combine both of them, which normalize the confidence over all related topics, i.e.,

$$\text{AverageConfidence}(q_j) = \frac{1}{m} \sum_{v_i \in V(q_j)} \frac{|T(q_j) \cap T(v_i)|}{|T(q_j)|} \quad (8)$$

We finally choose the average confidence as the fourth constraint in our optimization framework.

The last constraint is called *COI avoidance*. In many cases, we need to consider the conflict-of-interest (COI) problem. For example, an author, of course, should not review his own or his coauthors' paper. This can be accomplished through employing a binary $M \times N$ matrix U . An element with value of 0, i.e., $U_{ij} = 0$, represents expert v_i has the conflict-of-interest with query q_j . A simple way is to multiply the matrix U with the matching score R in (Eq.1).

Finally, by incorporating different constraints in Eq. 4-8 and the COI matrix U into the basic objective function (Eq. 1), we can result in a constrain-based optimization framework, e.g.:

$$\begin{aligned} \text{Max} \quad & \sum_{v_i \in V} \sum_{q_j \in Q(v_i)} U_{ij} R_{ij} - \sum_{k=1}^K (\mu_k \sum_{j=1}^N |V^k \cap V(q_j)|^2) \\ & - \beta \sum_{v_i \in V} |Q(v_i)|^2 + \lambda \sum_{q_j \in Q} \sum_{v_i \in V(q_j)} \frac{|T(q_j) \cap T(v_i)|}{|T(q_j)|} \\ \text{s.t.} \quad & \forall q_j \in Q, |V(q_j)| = m \\ & \forall v_i \in V, n_1 \leq |Q(v_i)| \leq n_2 \end{aligned} \quad (9)$$

where λ , β and μ_k are lagrangian multipliers, used to trade off the importance of different components in the objective function.

Now the problem is how to define the topic distribution θ , how to calculate the pairwise matching score R_{ij} , and how to optimize the framework.

4.2. Modeling Multiple Topics

The goal of topic modeling is to associate each expert v_i with a vector $\theta_{v_i} \in \mathbb{R}^T$ of T -dimensional topic distribution, and as well to associate each query q_j with a vector $\theta_{q_j} \in \mathbb{R}^T$. The topic distribution can be obtained in many different ways. For example, in the paper-reviewer assignment problem, each reviewer can select their expertise topics from a predefined categories. In addition, we can use statistical topic modeling [30, 31] to automatically extract topics from the input data. In this paper, we use the topic modeling approach to initialize the topic distribution of each expert and each query.

To extract the topic distribution, we can consider that we have a set of M expert documents and N query documents (each representing an expert or a query). An expert's document can be obtained by accumulating the content information related to the expert. For example, we combine all publication papers as the expert document of a reviewer, thus expert v_i 's document can be represented as $d_i = \{w_{ij}\}$. Each query can also be viewed as a document. Then we can learn these T topic aspects from the collection of expert documents and query documents using a topic model such as LDA [31]. Specifically, let $D = \{d_1, \dots, d_M\}$ be the set of experts' documents. The log-likelihood of the whole collection according to LDA is:

$$\log p(D|\theta, \phi) = \sum_{d \in D} \sum_{w \in d} c(w, d) \log \left(\sum_{z=1}^T p(w|z, \phi_z) p(z|d, \theta_d) \right) \quad (10)$$

where $c(w, d)$ is the count of word w in document d , $p(w|z, \phi_z)$ is the probability of topic z generating word w , and $p(z|d, \theta_d)$ is the probability of document d containing topic z .

We use the Gibbs sampling algorithm [32][33] to learn the topic distribution θ_{v_i} for each expert and each query. Once we obtain θ , we can represent each expert v_i by a topic distribution θ_{v_i} . Similarly, we can estimate a topic distribution θ_{q_j} for each query with an analogous model.

4.3. Pairwise Matching Score

We employ a language model-based retrieval method to calculate the pairwise matching score. With language model, the matching score R_{ij} between expert v_i and query q_j is interpreted as a probability $R_{ij}^{LM} = p(q_j|d_i) = \prod_{w \in q_j} p(w|d_i)$, where

$$p(w|d_i) = \frac{N_{d_i}}{N_{d_i} + \lambda_D} \cdot \frac{tf(w, d_i)}{N_{d_i}} + \left(1 - \frac{N_{d_i}}{N_{d_i} + \lambda_D}\right) \cdot \frac{tf(w, \mathbf{D})}{N_{\mathbf{D}}} \quad (11)$$

where N_{d_i} is the number of word tokens in document d_i , $tf(w, d_i)$ is the number of occurring times of word w in d_i , $N_{\mathbf{D}}$ is the number of word tokens in the entire collection, and $tf(w, \mathbf{D})$ is the number of occurring times of word w in the collection \mathbf{D} . λ_D is the Dirichlet smoothing factor and is commonly set according to the average document length in the collection [21].

Our previous work extended LDA and proposed the ACT model [34] to generate a topic distribution. By considering the learned topic model, we can define another matching score as

$$R_{ij}^{ACT} = p(q_j|d_i) = \prod_{w \in q_j} \sum_{z=1}^T p(w|z, \phi_z) p(z|d, \theta_{d_i}) \quad (12)$$

Further, we can define a hybrid matching score by combining the two probabilities together

$$R_{ij}^H = R_{ij}^{LM} \times R_{ij}^{ACT} \quad (13)$$

4.4. Optimization Solving

In order to maximize the objective function (Eq. 9), we construct a convex cost network with lower and upper bounds imposed on the arc flows. Figure 2 illustrates the constructing process as described in algorithm 1¹. Convex cost flow problem can be solved by transforming to an equivalent minimum cost flow problem[35]. The minimum cost flow of the network gives an optimal assignment with respect to (Eq. 9).

Theorem 1. *Algorithm 1 based on minimum convex cost flow gives an optimal solution.*

Proof. First the minimum convex cost flow problem (MCCF) can be formulated as the following optimization problem:

$$\begin{aligned} \text{Min} \quad & \sum_{(a,b) \in E(G)} C_{ab}(f(a,b)) \\ \text{s.t.} \quad & \forall a \in V(G), \sum_{b:(a,b) \in E(G)} f(a,b) = \sum_{b:(b,a) \in E(G)} f(b,a) \\ & \forall (a,b) \in E(G), l_{ab} \leq f(a,b) \leq u_{ab} \end{aligned} \quad (14)$$

¹Every arc in the network is associated with lower and upper bound denoted as $[l, u]$ and a convex function of the arc flow f .

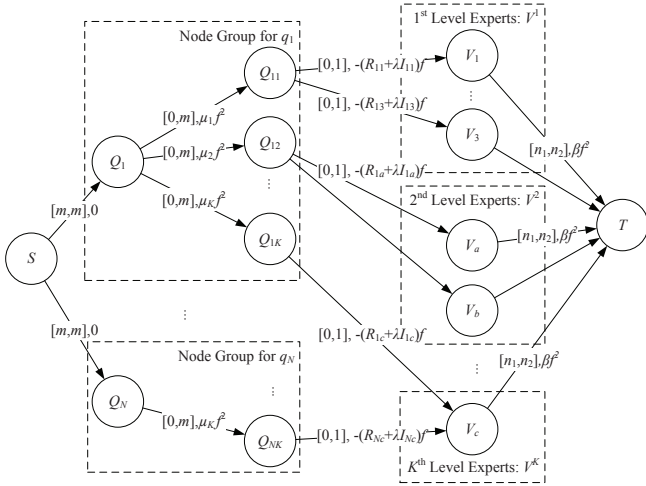


Figure 2: The construction of convex-cost network flow according to objective function 9.

The model is defined on a directed network $G = (V(G), E(G))$ with lower bound l_{ab} , upper bound u_{ab} and a convex cost function $C_{ab}(f(a, b))$ associated with every arc (a, b) .

Now we prove that minimizing (Eq. 14) on the graph G constructed in algorithm 1 is equivalent to maximizing (Eq. 9). For simplicity, we use I_{ij} to denote $\frac{|T(q_j) \cap T(v_i)|}{|T(q_j)|}$. For the constructing process, we see a feasible flow on G is mapping to a query-expert assignment. The flow from S to Q_j indicates the number of experts assigned with query q_j , and the flow from V_i to T indicates the number of queries assigned to expert v_i . And the cost between V_i and T is corresponding to the *load balance* soft penalty function (Eq. 4). The meaning of the flow from Q_j to Q_{jk} is the number of k th-level experts assigned to q_j , thus we impose a square cost function $\mu_k \cdot f^2$ on the arcs which is equivalent to the negative of the *authority balance* penalty. The flow from Q_{jk} to V_i means we assign query q_j to expert v_i , it is easy to find that no query will be assigned to the same expert twice since we give an upper bound of 1 on the arc, while the cost is equivalent to the negative of matching score and topic average confidence score. Therefore, our problem can be reduced to a equivalent MCF problem, where the objective function of MCF problem (Eq. 14) is the negative form of (Eq. 9). \square

In practice, it is not necessary to add all (Q_{jk}, V_i) arcs. To further reduce the complexity of the algorithm, we first greedily generate an assignment and preserve corresponding arcs, then keep only $c \cdot m$ arcs for Q_{jk} and $c \cdot n_2$ arcs for V_i which have highest matching score (c is a fixed constant). We call this process *Arc-Reduction*, which will reduce the number of arcs in the network without influencing the performance too much. To process large scale data, we can leverage the parallel implementation of convex cost flow [36].

4.5. Online Matching

After an automatic expertise matching process, the user may provide feedbacks. Typically, there are two types of user feedbacks: (1) pointing out a false match; (2) specifying a new

Algorithm 1: Optimization solving algorithm.

Input: The set of experts V ; the set of queries Q ; the matching score matrix $R_{M \times N}$; the COI matrix $U_{M \times N}$; Number of expertise level K ; m, n_1, n_2 as described above.

Output: An assignment of experts to queries maximizing objective function 9.

- 1.1 Create a network G with source node S and sink node T ;
- 1.2 **foreach** $q_j \in Q$ **do**
- 1.3 Create $K + 1$ nodes, denoted as $Q_j, Q_{j1}, \dots, Q_{jk}$ respectively;
- 1.4 Add an arc from source node S to node Q_j , with zero cost and flow constraint $[m, m]$;
- 1.5 Add an arc from node Q_j to Q_{jk} , with square cost function $\mu_k f^2$ and flow constraint $[0, m]$;
- 1.6 **foreach** $v_i \in V$ **do**
- 1.7 Create a node V_i ;
- 1.8 Add an arc from V_i to sink node T , with square cost function βf^2 and flow constraint $[n_1, n_2]$;
- 1.9 **foreach** $v_i \in V, q_j \in Q, s.t. U_{ij} = 1$ **do**
- 1.10 $k = \text{expert level of } v_i$;
- 1.11 Add an arc from Q_{jk} to V_i , with linear cost function $-(R_{ij} - \lambda I_{ij})f$ and flow constraint $[0, 1]$;
- 1.12 Compute the minimum cost flow on G ;
- 1.13 **foreach** $v_i \in V, q_j \in Q, s.t. U_{ij} = 1$ **do**
- 1.14 $k = \text{expert level of } v_i$;
- 1.15 **if** $\text{flow } f(Q_{jk}, V_i) = 1$ **then** Assign query q_j to expert v_i ;

Algorithm 2: Online matching algorithm.

Input: A minimum cost network flow f on G corresponding to the current assignment; an inappropriate match (v_i, q_j) .

Output: A new assignment.

- 2.1 $k = \text{expert level of } v_i$;
- 2.2 **if** $f(Q_{jk}, V_i) = 1$ **then**
- 2.3 Construct the residual network $G(f)$;
- 2.4 Compute the shortest path P_{back} from T to S on $G(f)$ which contains backward arc (V_i, Q_{jk}) ;
- 2.5 Cancel(roll back) 1 unit of flow along P_{back} and update $G(f)$;
- 2.6 Remove arc (Q_{jk}, V_i) from G and update $G(f)$;
- 2.7 Compute shortest augmenting path P_{aug} from S to T ;
- 2.8 Augment 1 unit of flow along P_{aug} ;

match. Online matching aims to adjust the matching result according to the user feedback. One important requirement is how to perform the adjustment at real time. In our framework, we provide online interactive adjustment without recalculating the whole cost flow. For both types of feedbacks, we can easily accomplish online adjustment by canceling some flows and augmenting new assignments in our framework. We give algorithm 2 to consider the first type of feedback, which still produces an optimal solution.

Lemma 1 (Negative Cycle Optimality Conditions). [35] *A feasible solution f^* is an optimal solution of the minimum cost flow problem if and only if it satisfies the negative cycle optimality conditions: namely, the residual network $G(f^*)$ contains no negative cost cycle.*

Theorem 2. *Algorithm 2 produces an optimal solution in the network without assignment (q_j, v_i) .*

Proof. According to Lemma 1, the residual network $G(f)$ contains no negative cost cycle since the given flow f has the min-

imum cost. In algorithm 2, we remove the inappropriate match (v_i, q_j) and adjust the network flow in line 2.3-2.5. Denote the feasible flow in the network after line 2.5 as f' . According to the SAP (Short Augmenting Path) algorithm of cost flow, if f' has the minimum cost (i.e., $G(f')$ contains no negative cycle), the algorithm will give the optimal solution. We show the optimality of f' by contradiction. Assume $G(f')$ contains a negative cycle C , C must intersect with the shortest path P_{back} computed on line 2.3, since the original $G(f)$ contains no negative cycle. Thus merging C into path P_{back} will generate a shorter path, which contradicts with the assumption that P_{back} is shortest. Therefore, f' has the minimum cost. Accordingly, algorithm 2 gives the optimal solution after augmenting the new assignment. \square

5. Experimental Results

The proposed approach for expertise matching is very general and can be applied to many application to align experts and queries. We evaluate the proposed framework on two different genres of expertise matching problems: paper-reviewer assignment and course-teacher assignment. Three experiments on different datasets are conducted to show the effectiveness of the proposed method. All data sets, code, and detailed results are publicly available ². All the experiments are carried out on a PC running Windows XP with Intel Core2 Quad CPU Q9550(2.83GHz), 3.2G RAM.

5.1. Paper-Reviewer Assignment Experiment

Dataset The paper-reviewer dataset consists of 338 papers and 354 reviewers. The reviewers are program committee members of KDD'09 and the 338 papers are those published on KDD'08, KDD'09, and ICDM'09. For each reviewer, we collect his/her all publications from an academic search system Arnetminer³[37] to generate the expertise document. As for the COI problem, we generate the COI matrix U according to the coauthor relationship in the last five years and the organization they belong to. Finally, we set that a paper should be reviewed by $m = 5$ experts, and an expert at most reviews $n_2 = 10$ papers.

Baseline Methods and Evaluation Metrics We employ a greedy algorithm as the baseline. The greedy algorithm assigns experts with highest matching score to each query, while keeping the load balance for each expert (i.e., $|Q(v_i)| \leq n_2$) and avoiding the conflict of interest.

As there are no standard answers, in order to quantitatively evaluate our method, we define the following metrics:

Matching Score (MS): It is defined as the accumulative matching score.

$$MS = \sum_{v_i \in V} \sum_{q_j \in Q(v_i)} U_{ij} R_{ij}$$

²<http://www.arnetminer.com/expertisematching>

³<http://arnetminer.org>

Table 2: Effects of different constraints on matching score.

| Constraint | Matching Score |
|--|----------------|
| Basic objective function (Eq. 1) | 635.51 |
| + Load Balance <i>soft penalty</i> with $\beta = 0.02$ | 592.83 |
| + Authority Balance with $\mu = (0.02, 0)^T$ | 599.37 |
| + COI | 590.14 |

Load Variance (LV): It is defined as the variance of the number of papers assigned to different reviewers.

$$LV = \sum_{i=1}^M \left(|Q(v_i)| - \frac{\sum_{i=1}^M |Q(v_i)|}{M} \right)^2$$

Expertise Variance (EV): It is defined as the variance of the number of top level reviewers assigned to different papers.

$$EV = \sum_{j=1}^N \left(|V(q_j) \cap V^1| - \frac{\sum_{j=1}^N |V(q_j) \cap V^1|}{N} \right)^2$$

Results In this experiment, we tune different parameters to analyze the influence on the accumulative matching score. We also evaluate the efficiency of our proposed approach.

We first set $\mu = \mathbf{0}$ and tune the parameter β to find out the effects of soft penalty function. Figure 3 (a) illustrates how soft penalty function influences the matching score with different β . We see that the matching score decreases slightly with β increasing. Figure 3 (b) shows the effects of load variance with β varied. We see that the load variance changes very fast toward balance.

In figure 3 (c), we compare the two different methods to achieve load balance, namely, strict constraint and soft penalty. The two LV-MS curves are respectively generated by setting different minimum numbers n_1 for strict constraint and varying the weight parameter β for soft load balance penalty. The curves show that soft penalty outperforms strict constraint towards load balance.

Then we set β to 0 to test the effects of authority balance. Experts are divided into 2 levels base on their H-index, and we set $\mu_2 = 0$ to consider the balance of the senior reviewers only. Figure 4 presents the accumulative matching score (a) and expertise variance (b) with μ_1 varied.

Further, we analyze the effects of different constraints. Specifically, we first remove all constraints (using Eq. (1) only), and then add the constraints one by one in the order (Load balance, Authority balance, and COI). In each step, we perform expertise matching using our approach. Table 2 lists the accumulative matching score obtained in each step. We see that the load balance constraint will reduce the expertise matching score, while the other constraints have little negative effect. This is because senior experts are often good at many aspects, thus assigned with heavy load in traditional matching. In our approach the decrease of matching score in the load balance constraint is to balance the work load of senior experts.

Finally, we evaluate the efficiency performance of the proposed algorithm. We compare the CPU time of the original optimal algorithm and the version with *Arc-Reduction*. As shown

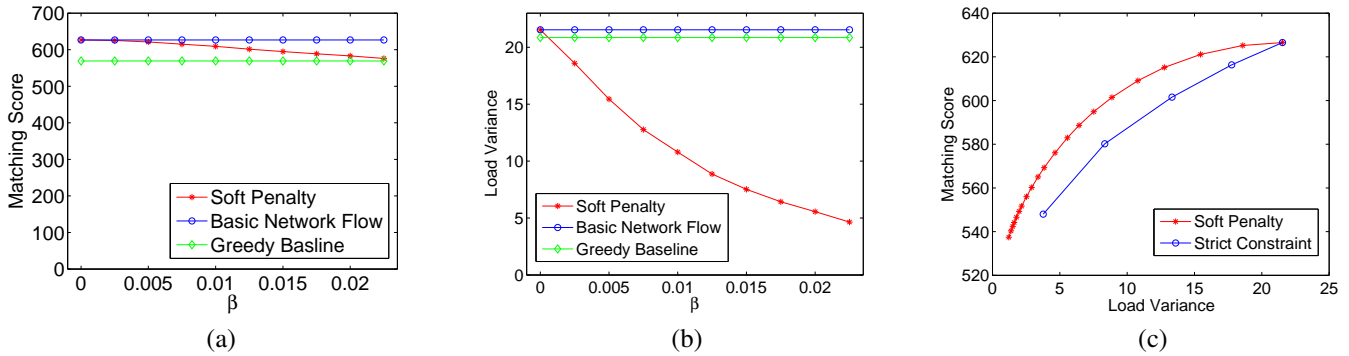


Figure 3: Figure (a) and (b) illustrate how soft penalty function influences the matching score (MS) and load variance with different β respectively. Figure (c) gives a comparison between soft penalty function and strict constraint methods towards load balance.

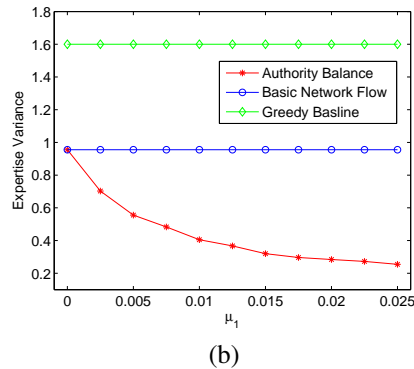
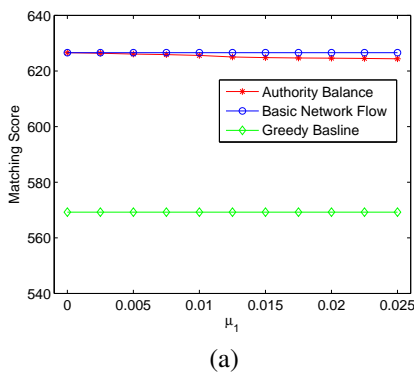


Figure 4: Matching score (MS) and expertise variance (EV) with μ_1 varied.

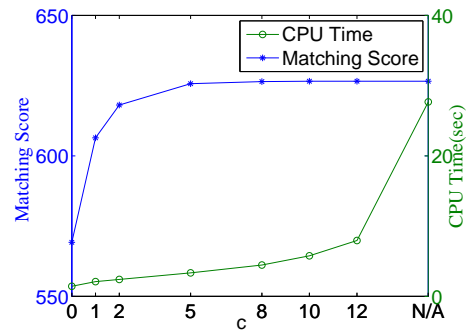


Figure 5: Efficiency performance (s).

| Reviewer | Assigned papers |
|-------------|---|
| Lise Getoor | Evaluating Statistical Tests for Within-Network Classifiers of ... Discovering Organizational Structure in Dynamic Social Network Connections between the lines: augmenting social networks with text MetaFac: community discovery via relational hypergraph factorization Relational learning via latent social dimensions Influence and Correlation in Social Networks |
| Wei Fan | Mining Data Streams with Labeled and Unlabeled Training Examples Vague One-Class Learning for Data Streams Active Selection of Sensor Sites in Remote Sensing Applications Name-ethnicity classification from open sources Consensus group stable feature selection Categorizing and mining concept drifting data streams |
| Jie Tang | Co-evolution of social and affiliation networks Influence and Correlation in Social Networks Feedback Effects between Similarity and Social Influence ... Mobile call graphs : beyond power-law and lognormal distributions Audience selection for on-line brand advertising: privacy-friendly ... |

Table 3: Example assigned papers to three reviewers.

5.2. Multi-topic Paper-Reviewer Assignment Experiment

in Figure 5, the *Arc-Reduction* process can significantly reduce the time consumption. For example, when setting $c = 12$ in this problem, we can achieve a $> 3\times$ speedup without any loss in matching score.

We further use a case study (as shown in table 5.1 and 4) to demonstrate the effectiveness of our approach. We see that the result is reasonable. For example, Lise Getoor, whose research interests include relational learning, is assigned with a lot of papers about social network.

Dataset We use another dataset (D2) to verify the performance on “topic coverage”. The second dataset D2 is provided by [3]⁴, consisting of 73 queries and 189 reviewers. The 73 queries are paper abstracts from SIGIR’07, where each of them is related to at least two topics. The documents of reviewers are the combination of all abstracts published in SIGIR1971-2006 by the corresponding reviewers. In this dataset, 25 topics such as “text mining”, “clustering” and “language models” are identified by

⁴<http://timan.cs.uiuc.edu/data/review.html>

| Paper | Assigned reviewers |
|---|---|
| Audience selection for on-line brand advertising: privacy-friendly social network targeting | C. Lee Giles, Jie Tang, Matthew Richardson, Hady Wirawan Lauw, Elena Zheleva |
| Partitioned Logistic Regression for Spam Filtering | Rong Jin, Chengxiang Zhai, Saharon Rosset, Masashi Sugiyama, Annalisa Appice |
| Structured Learning for Non-Smooth Ranking Losses | Xian-sheng Hua, Tie-yan Liu, Hang Li, Yunbo Cao, Lorenza Saitta |
| Unsupervised deduplication using cross-field dependencies | Chengxiang Zhai, Deepak Agarwal, Max Welling, Donald Metzler, Oren Kurland |
| The structure of information pathways in a social communication network | C. Lee Giles, Wolfgang Nejdl, Melanie Gnasa, Michalis Faloutsos, Cameron Marlow |

Table 4: List of reviewers for 5 random papers.

human experts, and the related topics of each query and each reviewer are manually labeled as a gold standard for evaluation. We use the Lemur toolkit [38] for pre-processing to tokenize each query and document, removing common stop words.

Comparison Methods We implement the baseline methods in [3]. The baseline approach use a language model to retrieve reviewer documents for each query. Specifically, given a query q_j , one way to rank reviewers is to use the probability of query q_j given reviewer document d_i , i.e.,

$$p(q_j|d_i) = \prod_{w \in q_j} \frac{tf(w, d_i) + \mu p(w|D)}{N_{d_i} + \mu} \quad (15)$$

where N_{d_i} is the document size of d_i , $tf(w, d_i)$ is the number of occurring times of word w in d_i , $p(w|D)$ is the unigram language model of the entire reviewer document collection and μ is a Dirichlet smoothing factor which we empirically set to 1,000 according to [3]. In addition, we use a smoothed version of $p(w|D)$ so it would not return zero when word w is unseen in the entire document collection, i.e.,

$$p(w|D) = \frac{tf(w, D) + 1}{N_D + V} \quad (16)$$

where V is the estimated vocabulary size (i.e. the number of distinct words).

We also implement the baseline method that use KL-divergence for reviewer retrieval [21], and the final formula is as follows,

$$\sum_{w \in q_j, tf(w, d_i) > 0} p(w|q_j) \log(1 + \frac{tf(w, d_i)}{\mu p(w|D)}) + \log \frac{\mu}{\mu + N_{d_i}} \quad (17)$$

The two baseline methods will be referred as Baseline-Pr and Baseline-KL in following discussion.

Setting For a fair comparison with the different methods in [3], we change some constraints so that all methods can be compared in same constraint settings. Specifically, for this dataset we remove the COI constraint, set that the paper should be reviewed by $m = 3$ experts, and we test our framework with and without expert load balance constraint. In addition, we implement our framework which removes authority balance constraint, containing *Coverage* score and pairwise matching score R_{ij} in our objective function. The new construction of network flow is shown in Figure 6. We select top-k topics as related topics of queries and experts. $Q_{j1} \dots Q_{jk}$ are nodes of related topics for query q_j , σ is a lagrangian multiplier used to adjust the importance of *Coverage*, and θ_{jk} are probabilities of related topics given query q_j . The setting of parameter k will be discussed in following subsections. Finally, we use *Coverage* (Eq. 6), *Confidence* (Eq. 7), *AverageConfidence* (Eq. 8), and *F_score* as the measures to quantitatively compare the different methods.

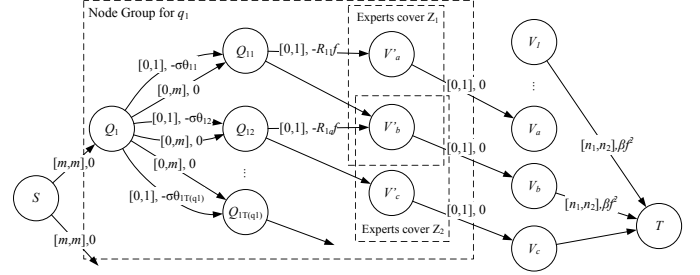


Figure 6: The construction of network flow for dataset D2. The network has been simplified for a clear view.

Results We compare the results of different methods on dataset D2. All methods are well-tuned and the PLSA method is the best one proposed in [3]. Results are presented in Table 5, where all measures are averaged for 73 queries.

From Table 5 we can make meaningful observations. First, all methods outperform the two baseline methods on *Coverage*, *AverageConfidence* and *F_score*. The baseline achieved better *Confidence* because assigned reviewers are more likely to cover one same topic for a query. As a result, the PLSA and our method are better than traditional IR methods for reviewer-assignment problem. Second, our method Flow achieved comparable performance with PLSA method under same constraints. However, it is even worth to mention that our framework can achieve good results when load balance constraint is added, because in real reviewer assignment problems, load balance is an important factor that must be considered. Flow-L5 or Flow-L10 is the framework that allows 5 or 10 queries at most for each reviewer, and from Table 5 it is clear that they still outperforms the baseline.

Parameter Sensitivity Now we show the analysis of different parameters in our framework. This is necessary to test the robustness of our framework, and also tells us how to set these parameters for good performance.

The first parameter we test is the lagrangian multiplier σ which controls the importance of *Coverage* score. Figure 7 shows the performance is not very sensitive to this parameter. When σ is set to relatively large values, both *Coverage* and *Confidence* become stable. Moreover, it can be seen that when only consider pairwise matching scores R_{ij} (i.e. $\sigma = 0$), both *Coverage* and *Confidence* are low, which is consistent with the baseline methods. We also got similar results when only consider *Coverage* scores. As a result, the experiment indicates that a hybrid framework can achieve better performance when appropriately sets component importance.

The number of topics also affects the performance. We apply the Gibbs sampling algorithm to learn topic models for different

| Methods | Coverage | Confidence | AverageConfidence | F _{score} |
|-------------|----------|------------|-------------------|--------------------|
| Baseline-Pr | 74% | 62% | 46% | 63% |
| Baseline-KL | 75% | 62% | 45% | 63% |
| PLSA | 87% | 58% | 53% | 67% |
| Flow | 87% | 60% | 51% | 67% |
| Flow-L10 | 86% | 58% | 49% | 66% |
| Flow-L5 | 80% | 59% | 48% | 65% |

Table 5: Comparison of all methods on different measures. The number of topics was set to 20 when learning topic model.

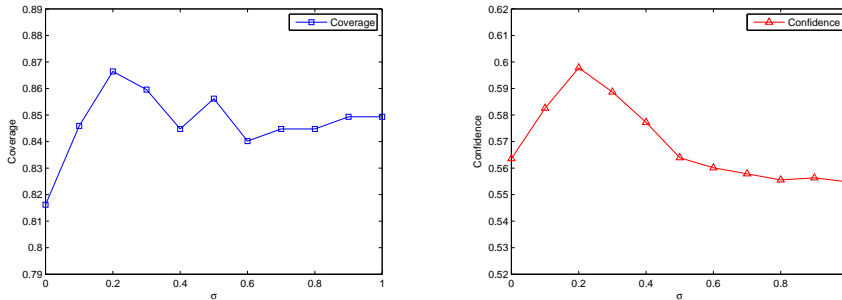


Figure 7: Sensitivity of Coverage and Confidence to different lagrangian multiplier σ .

number of topics (e.g. 10, 30, 50 topics). In addition, we vary the number of related topics (i.e. $T(q_j)$ and $T(v_i)$) from 1 to 7, and $T(q_j)$ $T(v_i)$ are determined by selecting top-k topics in θ_{q_j} and θ_{v_i} as we have discussed in Section 4.1. Finally the sensitivity curves of Coverage and Confidence are plotted in Figure 8. With the help of pairwise matching scores, Coverage and Confidence remain $> 80\%$ and $> 54\%$ even we set $k = 1$ and topic number to 10. Moreover, we see that relatively large and relatively small number of related topics do not give good results. The appropriate number of related topics is about 3 or 4, which is fairly near the gold standard. Setting too few topics (e.g. 10) may generally hurt the final performance, but using enough topics (20, 30, 50) would not make big difference.

5.3. Course-Teacher Assignment Experiment

Dataset In the course-teacher assignment experiment, we manually crawled graduate courses from the department of Computer Science (CS) of four top universities, namely CMU, UIUC, Stanford, and MIT. In total, there are 609 graduate courses from the fall semester in 2008 to 2010 spring, and each course is instructed by 1 to 3 teachers. Our intuition is that teachers’ research interest often match the graduate courses he/she is teaching. Thus we still use the teachers’ recent (five years) publications as their expertise documents, while the course description and course name are taken as the query.

Baseline Methods and Evaluation Metrics We employ the same greedy method used in experiment 5.1 as baseline. The real assignment is extracted as the ground-truth. Thus, we perform the evaluation in terms of precision.

Results Figure 9 (a) shows the assignment precision in the course-teacher assignment task by our approach and the baseline method, and (b) shows the effects of the parameter β on the precision on UIUC data. The precision is defined as the

ratio of the number of correct assignments (consistent with the ground truth data) over total number of assignments. As Figure 9 (a) shows, in all the data sets we collect from top universities, our algorithm outperforms the greedy method greatly. And in Figure 9 (b), as the β increases, the precision of our approach increases in general and decreases slowly after it exceeds the peak value. The peak value is more than 50 percents larger than the initial precision, which validates the effectiveness of the soft penalty approach.

We conduct a further analysis on the UIUC data set. As Table 6 shows, some professors with publications in various domains, are likely to be assigned with many courses in the baseline algorithm. But in real situation, most professors, though with various background, want to focus on several directions. Thus some courses should be assigned to younger teachers. While in our algorithm, the situation is much better. And we can see that each teacher is assigned with a reasonable load as well as a centralized interest.

5.4. Online System

Based on the proposed method, we have developed an online system for paper-reviewer suggestions, which is available at ⁵. Figure 10 shows a screenshot of the system. The input is a list of papers (with titles, abstracts, authors, and organization of each author) and a list of conference program committee (PC) members. We use the academic information stored in Arnet-Miner to find the topic distribution for each paper and each PC member [37]. With the two input lists and the topic distribution, the system automatically finds the match between papers and authors. As shown in Figure 10, there are 5-7 papers assigned to each PC member and the number of reviewers for each paper

⁵<http://review.arnetminer.org/>

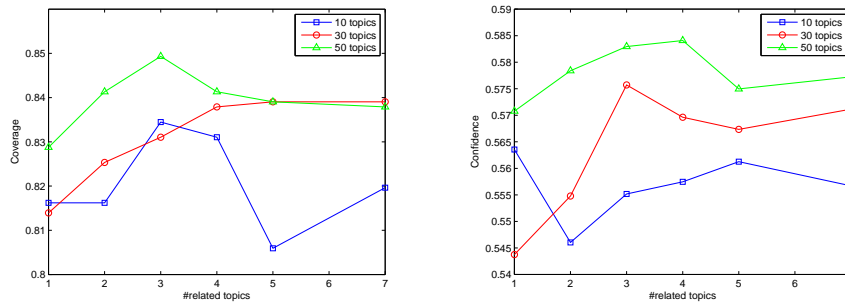


Figure 8: Sensitivity of *Coverage* and *Confidence* to the number of related topics for different topic models, with $\sigma = 0.3$.

| Professor | Pub Papers | Courses assigned(baseline) | Courses assigned(our approach) |
|-----------------|------------|---|--|
| Jose Meseguer | 237 | 23 courses Database Systems (2008,spring) Programming Languages and Compilers (2008,spring) Iterative and Multigrid Methods (2009,spring) Programming Languages and Compilers (2009,spring) | 7 courses Programming Languages and Compilers (2008,spring) Programming Language Semantics (2008,spring) Programming Languages and Compilers (2008,fall) Programming Languages and Compilers (2009,spring) |
| ChengXiang Zhai | 117 | 18 courses Computer Vision (2009,spring) Text Information Systems (2009,spring) Stochastic Processes and Applic (2009,fall) Computer Vision (2008,spring) | 7 courses Text Information Systems (2008,spring) Stochastic Processes and Applic (2008,fall) Text Information Systems (2009,spring) Stochastic Processes and Applic (2009,fall) |

Table 6: Case study: professors with many courses assigned in UIUC(2008, fall - 2010, spring)

is set as 3. The system will also avoid the conflict-of-interest (COI) according to the coauthorship and co-organization relationship. In addition, users can provide feedbacks for online adjustment, by removing or confirm (fix) an assignment.

6. Conclusion and Future Work

In this paper, we study the problem of expertise matching in a constrain-based framework. We formalize the problem as a minimum convex cost flow problem. We theoretically prove that the proposed approach can achieve an optimal solution and develop an efficient algorithm to solve it. Experimental results on two different types of data sets demonstrate that the proposed approach can effectively and efficiently match experts with the queries. Also we provide an algorithm to consider user feedbacks in real time. We are now applying the proposed method to several real-world applications. Feedbacks from the users are very positive.

The general problem of expertise matching represents an new and interesting new research direction. There are many potential future directions of this work. One interesting issue is to apply the proposed framework to question answer (e.g., Yahoo! Answer), where one of the most important issues is how to identify who can answer a new question. Another interesting issue is to incorporate some supervised information into our framework to further improve the performance of expertise matching. Finally, it is important to consider the influence between users when extending expertise matching to the social network.

ACKNOWLEDGMENTS

The work is supported by the Natural Science Foundation of China (No. 61073073 , No. 60703059 , No. 60973102),

Chinese National Key Foundation Research (No. 60933013), National High-tech R&D Program (No. 2009AA01Z138). The work was done when the last author is visiting Tsinghua University.

References

- [1] C. B. Haym, H. Hirsh, W. W. Cohen, C. Nevill-manning, Recommending papers by mining the web, in: Proceedings of the 20th international joint conference on Artificial intelligence (IJCAI'99), 1999, pp. 1–11.
- [2] S. T. Dumais, J. Nielsen, Automating the assignment of submitted manuscripts to reviewers, in: SIGIR '92: Proceedings of the 15th annual international ACM SIGIR conference on Research and development in information retrieval, ACM, New York, NY, USA, 1992, pp. 233–244. doi:http://doi.acm.org/10.1145/133160.133205.
- [3] M. Karimzadehgan, C. Zhai, G. Belford, Multi-aspect expertise matching for review assignment, in: Proceedings of the 17th ACM international conference on Information and knowledge management (CIKM'08), 2008, pp. 1113–1122.
- [4] D. Mimno, A. McCallum, Expertise modeling for matching papers with reviewers, in: Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (SIGKDD'07), 2007, pp. 500–509.
- [5] M. Karimzadehgan, C. Zhai, Constrained multi-aspect expertise matching for committee review assignment, in: Proceedings of the 17th ACM international conference on Information and knowledge management (CIKM'09), 2009, pp. 1697–1700.
- [6] D. Hartvigsen, J. C. Wei, R. Czuchlewski, The conference paper-reviewer assignment problem, Decision Sciences 30 (3) (1999) 865–876.
- [7] Y.-H. Sun, J. Ma, Z.-P. Fan, J. Wang, A hybrid knowledge and model approach for reviewer assignment, in: Proceedings of the 40th Hawaii International Conference on Systems Science (HICSS-40 2007), 2007, pp. 47–47.
- [8] S. Hettich, M. J. Pazzani, Mining for proposal reviewers: lessons learned at the national science foundation, in: Proceeding of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD'06), 2006, pp. 862–871.
- [9] D. Conry, Y. Koren, N. Ramakrishnan, Recommender systems for the conference paper assignment problem, in: RecSys '09: Proceedings of the third ACM conference on Recommender Systems, 2009, pp. 1–11.

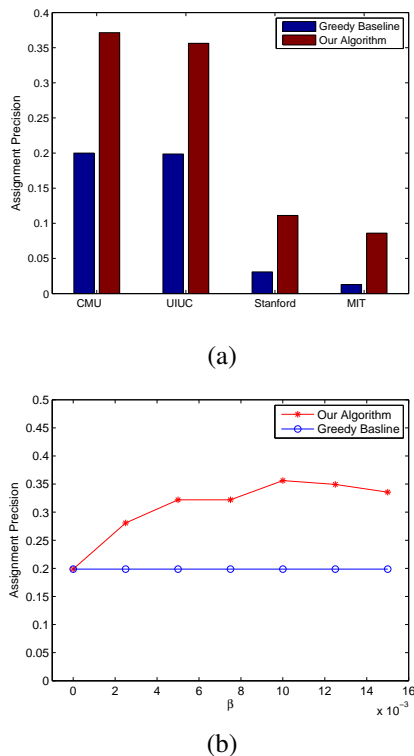


Figure 9: Course-Teacher Assignment performance(%).

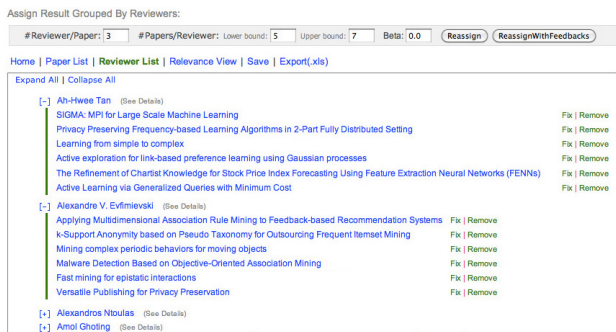


Figure 10: Screenshot of the online system.

mender systems, ACM, New York, NY, USA, 2009, pp. 357–360. doi:<http://doi.acm.org/10.1145/1639714.1639787>.

[10] N. D. Mauro, T. M. A. Basile, S. Ferilli, Grape: An expert review assignment component for scientific conference management systems, in: IEA/AIE'05: Proceedings of the 18th International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems, 2005, pp. 789–798.

[11] H. Fang, C. Zhai, Probabilistic models for expert finding, in: Proceedings of the 29th European Conference on Information Retrieval Research (ECIR'07), 2007, pp. 418–430.

[12] D. Petkova, W. B. Croft, Hierarchical language models for expert finding in enterprise corpora, International Journal on Artificial Intelligence Tools (2008) 5–18.

[13] K. Balog, L. Azzopardi, M. de Rijke, Formal models for expert finding in enterprise corpora, in: Proceedings of the 29th ACM SIGIR International Conference on Information Retrieval (SIGIR'2006), 2006, pp. 43–55.

[14] W. Tang, J. Tang, C. Tan, Expertise matching via constraint-based optimization, in: In Proceedings of 2010 IEEE/WIC/ACM International Conferences on Web Intelligence (WI'2010), 2010.

[15] D. Yimam, A. Kobsa, Demoir: A hybrid architecture for expertise model-

ing and recommender systems, in: In Proceedings of the Ninth IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises, 2000, pp. 67–74.

[16] Y. Cao, J. Liu, S. Bao, H. Li, Research on expert search at enterprise track of trec 2005, in: TREC, 2005.

[17] Y. Fu, W. Yu, Y. Li, Y. Liu, M. Zhang, S. Ma, Thuir at trec 2005: Enterprise track (2005).

[18] C. Basu, H. Hirsh, W. W. Cohen, C. Nevill-Manning, Technical paper recommendation: a study in combining multiple information sources, J. Artif. Int. Res. 14 (2001) 231–252.

[19] C. Basu, H. Hirsh, W. Cohen, Recommendation as classification: Using social and content-based information in recommendation, in: In Proceedings of the Fifteenth National Conference on Artificial Intelligence, AAAI Press, 1998, pp. 714–720.

[20] D. Yarowsky, R. Florian, Taking the load off the conference chairs: Towards a digital paper-routing assistant (1999).

[21] C. Zhai, J. Lafferty, A study of smoothing methods for language models applied to ad hoc information retrieval, in: Proceedings of the 24th ACM SIGIR International Conference on Information Retrieval (SIGIR'01), 2001, pp. 334–342.

[22] K. Balog, L. Azzopardi, M. de Rijke, A language modeling framework for expert finding (2008).

[23] X. Wei, W. B. Croft, Lda-based document models for ad-hoc retrieval, in: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval, SIGIR'06, ACM, New York, NY, USA, 2006, pp. 178–185.

[24] J. J. M. Guervós, P. A. C. Valdivieso, Conference paper assignment using a combined greedy/evolutionary algorithm, in: PPSN, 2004, pp. 602–611.

[25] E. Zitzler, L. Thiele, Multiobjective evolutionary algorithms: A comparative case study and the strength pareto approach (1999).

[26] M. Cámara, J. Ortega, F. de Toro, A single front genetic algorithm for parallel multi-objective optimization in dynamic environments, Neurocomput. 72 (2009) 3570–3579.

[27] C. J. Taylor, On the optimal assignment of conference papers to reviewers, Tech. Rep. MS-CIS-08-30, Computer and Information Science Department, University of Pennsylvania (2008).

[28] M. A. Rodriguez, J. Bollen, An algorithm to determine peer-reviewers, in: Proceedings of the Conference on Information and Knowledge Management, ACM Press, Napa, California, 2008, pp. 319–328. doi:10.1145/1458082.1458127.

[29] S. Benferhat, J. Lang, Conference paper assignment, Int. J. Intell. Syst. 16 (10) (2001) 1183–1192.

[30] T. Hofmann, Probabilistic latent semantic indexing, in: Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'99), 1999, pp. 50–57.

[31] D. M. Blei, A. Y. Ng, M. I. Jordan, Latent dirichlet allocation, Journal of Machine Learning Research 3 (2003) 993–1022.

[32] T. L. Griffiths, M. Steyvers, Finding scientific topics, in: Proceedings of the National Academy of Sciences (PNAS'04), 2004, pp. 5228–5235.

[33] J. Cao, T. Xia, J. Li, Y. Zhang, S. Tang, A density-based method for adaptive lda model selection, Neurocomput. 72 (2009) 1775–1781.

[34] J. Tang, R. Jin, J. Zhang, A topic modeling approach and its integration into the random walk framework for academic search, in: Proceedings of 2008 IEEE International Conference on Data Mining (ICDM'08), 2008, pp. 1055–1060.

[35] R. K. Ahuja, T. L. Magnanti, J. B. Orlin, Network Flows: Theory, Algorithms, and Applications, Prentice Hall, 1993.

[36] P. Beraldi, F. Guerriero, R. Musmanno, Parallel algorithms for solving the convex minimum cost flow problem, Computational Optimization and Applications 18 (2) (2001) 175–190.

[37] J. Tang, J. Zhang, L. Yao, J. Li, L. Zhang, Z. Su, Arnetminer: Extraction and mining of academic social networks, in: Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (SIGKDD'08), 2008, pp. 990–998.

[38] The lemur toolkit for language modeling and information retrieval. URL <http://lemurproject.org/>